

```

10      .OPT LIST
20      ::::::::::::::::::::::::::::::::::::::::::::
30      ;          QUICK MENU  -Final
40      ;          REV 3.2 (1984)
50      ;          by  Carlos Reyes
60      ::::::::::::::::::::::::::::::::::::::::::::
70      ;
80      ; ZERO PAGE VARIABLES
90      ;
=0043      0100 WORD = $43          ; STARTING ADDR OF BLOCK BEING READ
=0043      0110 COUNT = $43        ; # OF ENTRIES IN DIRECTORY
=0045      0120 WORD1 = $45        ; ENDING ADDR
=0047      0130 NXTSECT = $47      ; NEXT SECTOR TO BE READ FROM FILE
=0049      0140 XSAVE = $49        ; TEMP STORAGE FOR X REG
          0150 ;
          0160 ; OS EQUATES
          0170 ;
=0006      0180 TRAMSZ = 6
=0009      0190 BOOT? = 9
=0012      0200 RTCLOK = $12
=002A      0210 ICAX1Z = $2A
=002B      0220 ICAX2Z = $2B
=0058      0230 SAVMSC = 88
=0244      0240 COLDST = $0244
=02E0      0250 RUNAD = $02E0
=02E2      0260 INITAD = $02E2
=0301      0270 DUNIT = $0301
=0302      0280 DCOMND = $0302
=0303      0290 DSTATS = $0303
=0304      0300 DBUFLO = $0304
=0305      0310 DBUFHI = $0305
=030A      0320 DAUX1 = $030A
=030B      0330 DAUX2 = $030B
=03F8      0340 BASICFLG = $03F8 ; (??) TELLS THE OS TO SWITCH OUT BASIC
          0350 ; *NOTE: THIS IS ONLY VALID ON THE NEW XL SERIES
          0360 ;
=E400      0370 VECTORS = $E400
=E453      0380 DSKINV = $E453
=E459      0390 SIOV = $E459
=E474      0400 WARMSV = $E474
          0410 ;
          0420 ; BINARY HEADER
          0430 ;
0000      0440      *= $0700
          0450 ;
0700      0460 START
0700 0003      0470      .BYTE 0,3
0702 0007      0480      .WORD START,INIT
0704 0607
          0490 ;
          0500 ; INIT THE MENU
          0510 ;
0706      0520 INIT
0706 A001      0530      LDY #1
0708 8409      0540      STY BOOT?      ; DISK BOOTED OK
070A 8C0103     0550      STY DUNIT      ; DRIVE #1
070D 842B      0560      STY ICAX2Z     ; GR.1
070F 8C0B03     0570      STY DAUX2      ; ($169)
0712 8CF803     0580      STY BASICFLG   ; SWITCH OUT BASIC ON THE NEXT WARM START
          0590 ;
0715 88        0600      DEY

```

```

0716 8C4402      0610      STY COLDST      ;Y=0
                  0620      ;
0719 A90C        0630      LDA #0C
071B 852A        0640      STA ICAX1Z
                  0650      ;
071D A210        0660      LDX ##10      ;OPEN #6,12,1,"S:" or GR.1+16
071F 203808      0670      JSR USEVECT
                  0680      ;
0722 A559        0690      LDA SAVMSC+1 ;GET ADDR OF DLIST
0724 8DAD07      0700      STA DLPTR+2
                  0710      ;
0727 A968        0720      LDA # <361-1
0729 8D0A03      0730      STA DAUX1
                  0740      ;
072C A506        0750      LDA TRAMSZ      ;SHOW IF CARTRIDGE IS PRESENT
072E F004        0760      BEQ LOOP
0730 A94A        0770      LDA #10+64      ; "*"
0732 9158        0780      STA (SAVMSC),Y ;Y=1
                  0790      ;
                  0800      ;LOOP: READ THE DIRECTORY
                  0810      ;
0734            0820      LOOP
0734 A952        0830      LDA #'R
0736 A208        0840      LDX # >BUFFER
0738 A0B0        0850      LDY # <BUFFER
                  0860      ;
073A 202608      0870      JSR CALLDSK
073D A200        0880      LDX #0
                  0890      ;
                  0900      ;LOOP1: PROCESS NEXT ENTRY
                  0910      ;
073F            0920      LOOP1
073F BDB008      0930      LDA BUFFER,X ;TEST FLAG BYTE
0742 F048        0940      BEQ WAIT      ;NEVER USED
                  0950      ;
0744 303C        0960      BMI NEXT      ;DELETED
0746 4A          0970      LSR A
0747 B039        0980      BCS NEXT      ;OPEN FOR OUTPUT
                  0990      ;
                  1000      ;FOUND A GOOD ENTRY
                  1010      ;
0749 A443        1020      LDY COUNT      ;SAVE THE STARTING ADDR FOR LATER
074B BDB308      1030      LDA BUFFER+3,X
074E 998008      1040      STA SECTLO,Y
0751 BDB408      1050      LDA BUFFER+4,X
0754 999808      1060      STA SECTHI,Y
                  1070      ;
0757 98          1080      TYA          ;PRINT THE LETTER FOR THE ENTRY
0758 69A1        1090      ADC #33+128 ;ADD THE OFFSET (C=0)
075A A003        1100      LDY #3
075C 9158        1110      STA (SAVMSC),Y
                  1120      ;
075E C8          1130      INY
                  1140      ;
                  1150      ;LOOP2: MOVE FILENAME TO SCREEN
                  1160      ;
075F            1170      LOOP2
075F C8          1180      INY
0760 E8          1190      INX
0761 BDB408      1200      LDA BUFFER+5-1,X
0764 E91F        1210      SBC #32-1      ; (C=0)

```

```

0766 9158      1220      STA (SAVMSC),Y
0768 C00F      1230      CPY #5+11-1
076A 90F3      1240      BCC LOOP2
                  1250 ;
076C A558      1260      LDA SAVMSC
076E 6913      1270      ADC #20-1      ; CARRY IS SET FROM CPY
0770 8558      1280      STA SAVMSC
0772 9002      1290      BCC SKIP
0774 E659      1300      INC SAVMSC+1
                  1310 ;
0776           1320      SKIP
0776 E643      1330      INC COUNT
0778 A543      1340      LDA COUNT
077A C918      1350      CMP #24      ; GOT 24 ENTRIES ALREADY?
077C F00E      1360      BEQ WAIT      ; YES, EXIT
                  1370 ;
077E 8A        1380      TXA      ; EQUALIZE X REG
077F 69F5      1390      ADC #-11     ; CARRY IS CLEARED FROM 'CMP #24'
0781 AA        1400      TAX
                  1410 ;
                  1420 ; NEXT: POINT TO NEXT ENTRY IN DIRECTORY SECTOR
                  1430 ;
0782           1440      NEXT
0782 8A        1450      TXA
0783 18        1460      CLC
0784 6910      1470      ADC #16
0786 AA        1480      TAX
                  1490 ;
0787 0A        1500      ASL A      ; AT END OF SECTOR?
0788 90B5      1510      BCC LOOP1    ; NO, GET NEXT ENTRY
078A B0AB      1520      BCS LOOP    ; YES, GET NEXT DIR SECTOR
                  1530 ;
                  1540 ; WAIT: GET KEY FROM USER
                  1550 ;
078C           1560      WAIT
078C A224      1570      LDX ##20+4
078E 203808    1580      JSR USEVECT
                  1590 ;
0791 C91B      1600      CMP #27      ; ESCAPE KEY?
0793 F075      1610      BEQ COPYIT    ; YES, COPY MENU
                  1620 ;
0795 E941      1630      SBC #'A      ; IF KEY>27 THEN C=1, SO WE DON'T SET IT
0797 C543      1640      CMP COUNT
0799 B0F1      1650      BCS WAIT
                  1660 ;
                  1670 ; PREPARE TO LOAD IN THE FILE
                  1680 ;
079B AA        1690      TAX
079C BD8008    1700      LDA SECTLO,X ; GET STARTING ADDR FROM TABLES
079F 8547      1710      STA NXTSECT
07A1 BD9808    1720      LDA SECTHI,X
07A4 8548      1730      STA NXTSECT+1
                  1740 ;
07A6 8A        1750      TXA
07A7 F002      1760      BEQ DLPTR    ; HANDLE LMS PROBLEM
07A9 E8        1770      INX
07AA E8        1780      INX
07AB           1790      DLPTR
07AB FE63BD    1800      INC #BD60+3,X
                  1810 ;
07AE 205008    1820      JSR CALLSIO

```

```

1830 ;
07B1 CA      1840      DEX
1850 ;
07B2 2DB108  1860      AND BUFFER+1 ; IS THIS A BINARY LOADABLE FILE?
07B5 C9FF    1870      CMP #$FF
07B7 D06A    1880      BNE REBOOT ; NO, RUN THE MENU AGAIN
1890 ;
07B9 A514    1900      LDA RTCLOK+2 ; WAIT FOR VBLANK TO POINT TO THE SCREEN
07BB         1910      LOOP3
07BB C514    1920      CMP RTCLOK+2
07BD F0FC    1930      BEQ LOOP3
1940 ;
1950 ; DO THE ACTUAL BINARY LOAD
1960 ;
07BF         1970      PUTNXT
07BF 204108  1980      JSR GETBYTE ; GET THE STARTING ADDR FOR THE BLOCK
07C2 8543    1990      STA WORD
07C4 204108  2000      JSR GETBYTE
07C7 8544    2010      STA WORD+1
07C9 2543    2020      AND WORD
07CB C9FF    2030      CMP #$FF
07CD F0F0    2040      BEQ PUTNXT
2050 ;
07CF 204108  2060      JSR GETBYTE ; GET THE ENDING ADDRESS
07D2 8545    2070      STA WORD1
07D4 204108  2080      JSR GETBYTE
07D7 8546    2090      STA WORD1+1
2100 ;
2110 ; READ A BLOCK OF DATA
2120 ;
07D9         2130      RDBLK
07D9 204108  2140      JSR GETBYTE
07DC 9143    2150      STA (WORD),Y
2160 ;
07DE E643    2170      INC WORD
07E0 D002    2180      BNE SKIP1
07E2 E644    2190      INC WORD+1
2200 ;
07E4         2210      SKIP1
07E4 A545    2220      LDA WORD1
07E6 C543    2230      CMP WORD
07E8 A546    2240      LDA WORD1+1
07EA E544    2250      SBC WORD+1
07EC B0EB    2260      BCS RDBLK ; GO BACK IF <=
2270 ;
07EE ADE202  2280      LDA INITAD
07F1 0DE302  2290      ORA INITAD+1
07F4 F0C9    2300      BEQ PUTNXT
2310 ;
2320 ; INIT THE PROGRAM
2330 ;
07F6 8649    2340      STX XSAVE
07F8 200708  2350      JSR GOINIT
07FB A649    2360      LDX XSAVE
2370 ;
07FD A000    2380      LDY #0 ; CLEAR Y REG
07FF BCE202  2390      STY INITAD
0802 BCE302  2400      STY INITAD+1
0805 F0B8    2410      BEQ PUTNXT ; FORCED BRANCH
2420 ;
0807         2430      GOINIT

```

```

0807 6CE202      2440      JMP (INITAD)
                  2450      ;
                  2460      ; WRITE THE MENU TO THE DISK
                  2470      ;
080A             2480      COPYIT
080A 88          2490      DEY
080B 8C0A03      2500      STY DAUX1      ; Y=0
080E 8C0B03      2510      STY DAUX2
0811 A957        2520      LDA #'W
0813 A207        2530      LDX #7
0815 202608      2540      JSR CALLDSK
                  2550      ;
0818 A080        2560      LDY #$80
081A 202C08      2570      JSR CALLDSK2
                  2580      ;
081D A208        2590      LDX #8
081F 88          2600      DEY
0820 202908      2610      JSR CALLDSK1
                  2620      ;
0823             2630      REBOOT
0823 4C74E4      2640      JMP WARMSV
                  2650      ;
                  2660      ; CALL THE RESIDENT DISK INTERFACE
                  2670      ;
0826             2680      CALLDSK
0826 8D0203      2690      STA DCOMND
0829             2700      CALLDSK1
0829 8E0503      2710      STX DBUFHI
082C             2720      CALLDSK2
082C 8C0403      2730      STY DBUFLO
                  2740      ;
082F EE0A03      2750      INC DAUX1
                  2760      ;
0832             2770      AGAIN
0832 2053E4      2780      JSR DSKINV
0835 30FB        2790      BMI AGAIN
0837 60          2800      RTS
                  2810      ;
                  2820      ; USE THE RESIDENT HANDLERS
                  2830      ;
0838             2840      USEVECT
0838 BD01E4      2850      LDA VECTORS+1,X
083B 48          2860      PHA
083C BD00E4      2870      LDA VECTORS,X
083F 48          2880      PHA
0840 60          2890      RTS
                  2900      ;
                  2910      ; GET THE NEXT BYTE FROM THE FILE
                  2920      ;
0841             2930      GETBYTE
0841 E07D        2940      CPX #$7D      ; (# OF DATA BYTES IN SECTOR)
0843 F005        2950      BEQ GETSECT
                  2960      ;
0845             2970      GIVEBYTE
0845 BDB008      2980      LDA BUFFER,X
0848 E8          2990      INX
0849 60          3000      RTS
                  3010      ;
084A             3020      GETSECT
084A A547        3030      LDA NXTSECT ; AT THE EOF?
084C 0548        3040      ORA NXTSECT+1

```

```

084E F02D      3050      BEQ DONE      ; YES, RUN THE PROGRAM
                3060      ;
                3070      ; GET THE NEXT DATA SECTOR
                3080      ;
0850          3090 CALLSIO
0850 A940      3100      LDA ##40
0852 8D0303    3110      STA DSTATS
                3120      ;
0855 A547      3130      LDA NXTSECT
0857 8D0A03    3140      STA DAUX1
085A A548      3150      LDA NXTSECT+1
085C 8D0B03    3160      STA DAUX2
                3170      ;
085F 2059E4    3180      JSR SIOV
0862 30EC      3190      BMI CALLSIO ; IF ERROR, TRY AGAIN
                3200      ;
                3210      ; PROCESS THE LINK
                3220      ;
0864 AD2D09    3230      LDA BUFFER+125
0867 2903      3240      AND #3
0869 8548      3250      STA NXTSECT+1
                3260      ;
086B AD2E09    3270      LDA BUFFER+126
086E 8547      3280      STA NXTSECT
                3290      ;
0870 AD2F09    3300      LDA BUFFER+127
0873 297F      3310      AND ##7F
0875 8D4208    3320      STA GETBYTE+1
                3330      ;
0878 A200      3340      LDX #0
087A 88        3350      DEY
                3360      ;
087B F0C8      3370      BEQ GIVEBYTE ; FORCED BRANCH
                3380      ;
                3390      ; RUN THE PROGRAM
                3400      ;
087D          3410 DONE
087D 6CE002    3420      JMP (RUNAD)
                3430      ;
                3440      ; STARTING SECTOR TABLES
                3450      ;
0880          3460 SECTLO *= **24
0898          3470 SECTHI *= **24
                3480      ;
                3490      ; SECTOR BUFFER
                3500      ;
08B0          3510 BUFFER *= **128

```

```

3520      .OPT LIST

```

PAGE 2
SYMBOLS

0032 AGAIN	=03F8 BASICFLG	=0009 BOOT?	0080 BUFFER	0026 CALLDISK
0029 CALLDISK1	002C CALLDISK2	0050 CALLSIO	=0244 COLDST	000A COPYIT
=0043 COUNT	=030A DAUX1	=030B DAUX2	=0305 DBUFHI	=0304 DBUFLO
=0302 DCOMND	07AB DLPTR	007D DONE	=E453 DSKINV	=0303 DSTATS
=0301 DUNIT	0041 GETBYTE	004A GETSECT	0045 GIVEBYTE	0007 GOINIT
=002A ICAX1Z	=002B ICAX2Z	0706 INIT	=02E2 INITAD	0734 LOOP
073F LOOP1	075F LOOP2	07BB LOOP3	0782 NEXT	=0047 NXTSECT
07BF PUTNXT	07D9 RDBLK	0023 REBOOT	=0012 RTCLOCK	=02E0 RUNAD
=0058 SAVMSC	0098 SECTHI	0080 SECTLO	=E459 SIOV	0776 SKIP
07E4 SKIP1	0700 START	=0006 TRAMSZ	0038 USEVECT	=E400 VECTORS
07BC WAIT	=E474 WARMSV	=0043 WORD	=0045 WORD1	=0049 XSAVE